

# Learning Low-Complexity Autoregressive Models via Proximal Alternating Minimization

Fu Lin and Jie Chen

**Abstract**—We consider the estimation of the state transition matrix in vector autoregressive models when the time sequence data is limited but nonsequence steady-state data is abundant. To leverage both sources of data, we formulate the problem as the least-squares minimization regularized by a Lyapunov penalty. Explicit cardinality or rank constraints are imposed to reduce the complexity of the model. The resulting nonconvex, nonsmooth problem is solved by using the proximal alternating linearization method (PALM). An advantage of PALM is that for the problem under investigation, it is globally convergent to a critical point and the objective value is monotonically decreasing. Furthermore, the proximal operators therein can be efficiently computed by using explicit formulas. Our numerical experiments on synthetic and real-world data demonstrate the effectiveness of the developed approach. In particular, our approach is advantageous over gradient projection method for the cardinality and for the  $\ell_1$  constraints.

**Keywords:** Autoregressive models, Lyapunov penalty, steady-state data, proximal alternating linearized minimization (PALM)

## I. INTRODUCTION

Vector autoregressive (VAR) models are widely used in the analysis of linear interdependence in time series. A key step in building the autoregressive model is the identification of the state transition matrix. When time sequence data is readily available, the standard approach is to solve a least-squares problem. In several modern applications, however, the dimension of the model is significantly larger than the number of time sequence measurements, which makes the model unidentifiable through the simple least-squares. Such scenarios include, for example, tracking the progression of brain neurological diseases, because the number of comprehensive brain scans is limited due to the cost or medical concerns [1]. In inferring the gene expression networks, the number of genes is typically much larger than the number of measurements because of the intrusive nature of the measuring techniques [2]–[4].

In such situations, regularization is a typical rescue. Regularization techniques may introduce structures to the transition matrix. In particular, sparsity and low-rank structures are extensively studied, in part because of the convenience of convex optimization [1], [3], [5]–[11]. In [5], a sparse VAR model is found via Lasso for gene regulatory networks. In [10], the state transition matrix is decomposed into a sparse matrix and a low-rank matrix by using convex penalty functions. Other approaches based on convex optimization can be found in [1], [3], [6]–[9], [11]

In this work, we consider a regularization that takes advantage of additional data sources. When the VAR model is stable and the steady-state data are abundant, the steady-state data can be used to improve the estimation accuracy [1], [3], [4], [8], [12], [13]. In [1], the Lyapunov penalty is proposed to make use of the steady-state nonsequence data in conjunction with the standard least-squares estimator. In [3], the perturbed steady-state data is used to infer the sparse, stable gene expression models. In [4], both steady-state and temporal data are integrated for estimating gene regulatory networks. Other work that employs steady-state data for system identification includes [8], [12], [13].

Hence, we move one step further, leveraging the availability of limited time sequences and the abundant steady-state data, meanwhile imposing additional structural constraints to reduce the complexity of the model. The resulting estimator is a least-squares estimator, regularized by the Lyapunov penalty, and constrained by an explicit sparsity or low-rank requirement. That is, the state transition matrix is constrained to have a specific number of nonzeros or a specific rank. The identification problem then becomes nonconvex (due to the Lyapunov penalty and the low-complexity constraint) and nonsmooth (due to the low-complexity constraint). We propose solving the problem by the proximal alternating linearization method (PALM). An advantage of PALM is that it is globally convergent to a critical point and the objective value monotonically decreases. Moreover, the proximal operators therein admit closed-form expressions and therefore the implementation is particularly simple. It is also straightforward for PALM to handle stability constraints and other convex low-complexity constraints (e.g.,  $\ell_1$  constraint or nuclear norm constraint).

Our contributions can be summarized as follows. First, we propose the use of explicit cardinality or rank constraints for the identification of low-complexity VAR models. By doing so, one can directly control the number of nonzero elements or the rank of the transition matrix. Second, we reformulate the objective function that lends itself to a powerful PALM algorithm. In particular, we show that the proximal operators therein can be computed via closed-form expressions. Third, we prove that the problem formulation satisfies Lipschitz conditions and the Kurdyka-Lojasiewicz (KL) property. As a result, we establish the global convergence of our algorithm to a critical point. Finally, we demonstrate by numerical experiments that our method outperforms the gradient projection method, for both the cardinality constraint and the  $\ell_1$  constraint.

The presentation is organized as follows. In Section II we formulate the optimization problem for a low-complexity VAR model. In Section III we present the PALM algorithm and derive explicit formulas for the proximal operators. We show

F. Lin is with the Systems Department, United Technologies Research Center, 411 Silver Ln, East Hartford, CT 06118, USA. E-mail: linf@utrc.utc.com  
J. Chen is with IBM Thomas J. Watson Research Center, 1101 Kitchawan Road, Yorktown Heights, NY 10598, USA. E-mail: chenjie@us.ibm.com

convergence by establishing the Lipschitz conditions and the KL property in Section IV. In Section V, we demonstrate the effectiveness of PALM and compare it with gradient projection. Finally, we summarize and discuss future directions in Section VI.

## II. VAR MODEL IDENTIFICATION VIA LYAPUNOV PENALTY

Consider a  $p$ -dimensional vector autoregressive model:

$$x(t+1) = Ax(t) + \epsilon(t), \quad (1)$$

where  $x(t) \in \mathbb{R}^p$  is the state vector at time  $t$ ,  $A \in \mathbb{R}^{p \times p}$  is the state transition matrix, and  $\epsilon(t) \in \mathbb{R}^p$  is a zero-mean white stochastic process. We assume that the autoregressive model (1) is stable; that is, all eigenvalues of  $A$  have modulus less than one. Then, the state vector  $x(t)$  has a steady-state distribution whose covariance matrix  $P$  is determined by the discrete-time Lyapunov equation

$$APA^T + Q = P, \quad (2)$$

where  $Q \in \mathbb{R}^{p \times p}$  is the covariance matrix of  $\epsilon(t)$  and  $(\cdot)^T$  denotes the matrix transpose operation. Linear systems theory says that  $P$  is positive definite if and only if  $A$  is stable.

Our objective is to identify the state transition matrix  $A$ . For the convenience of developing optimization details, we use  $X$  to replace the unknown  $A$  in what follows. Given a set of  $n$  time sequence measurements of  $x(t)$ , the standard least-squares estimation amounts to

$$\underset{X \in \mathbb{R}^{p \times p}}{\text{minimize}} \quad \frac{1}{2} \|XC - D\|_F^2, \quad (3)$$

where  $C := [x(1), \dots, x(n-1)] \in \mathbb{R}^{p \times (n-1)}$ ,  $D := [x(2), \dots, x(n)] \in \mathbb{R}^{p \times (n-1)}$ , and  $\|\cdot\|_F$  denotes the Frobenius norm. When the number of time sequence data is less than the dimension of the states (i.e.,  $p > n-1$ ), infinitely many solutions exist for (3) and hence the state transition matrix is unidentifiable.

We are interested in the scenario when the time sequence data is scarce but the steady-state nonsequence data is readily available. This situation arises in several applications [1], [7], [8], [12], [14] including identification of cancer expression networks [7] and gene regulatory networks [14]. In this case, Huang and Schneider [1] propose to use the Lyapunov penalty

$$\|XPX^T + Q - P\|_F^2$$

as a regularization term. They show that the Lyapunov penalty in conjunction with least-squares may improve the accuracy of the estimation. Since the covariance matrix  $P$  is unknown in practice, we approximate it by using the sample covariance

$$S := \frac{1}{N-1} \sum_{i=1}^N (z^i - \bar{z})(z^i - \bar{z})^T \text{ with } \bar{z} := \frac{1}{N} \sum_{i=1}^N z^i,$$

where  $\{z^i\}_{i=1}^N$  is the steady-state nonsequence data. Then, the identification problem becomes

$$\underset{X \in \mathbb{R}^{p \times p}}{\text{minimize}} \quad \frac{1}{2} \|XC - D\|_F^2 + \frac{\rho}{2} \|XSX^T + Q - S\|_F^2, \quad (4)$$

where  $\rho$  is a positive coefficient that balances the estimation error between time sequence and nonsequence data.

### A. Stability Constraint

Since stability is a necessary condition for the use of Lyapunov penalty, it is natural to constrain the formulation (4) by stability. Let  $\tau(X)$  denote the spectral radius of  $X$ , that is,  $\tau(X) := \max\{|\lambda_i|\}_{i=1}^p$ . Since  $\tau(X) \leq \|X\|_2$  and since  $\|X\|_2 \leq \|X\|_F$ , we have

$$\tau^2(X) \leq \|X\|_2^2 = \|XX^T\|_2 \leq \|XX^T\|_F.$$

Thus, a stable VAR model can be obtained by solving the following problem:

$$\begin{aligned} & \underset{X \in \mathbb{R}^{p \times p}}{\text{minimize}} \quad \frac{1}{2} \|XC - D\|_F^2 + \frac{\rho}{2} \|XSX^T + Q - S\|_F^2 \\ & \text{subject to} \quad \|XX^T\|_F^2 < 1. \end{aligned} \quad (5)$$

Alternatively, one can incorporate the stability constraint in the cost function with a sufficiently large coefficient  $\mu \geq 0$

$$\begin{aligned} & \underset{X \in \mathbb{R}^{p \times p}}{\text{minimize}} \quad \frac{1}{2} \|XC - D\|_F^2 + \frac{\rho}{2} \|XSX^T + Q - S\|_F^2 \\ & \quad + \frac{\mu}{2} \|XX^T\|_F^2. \end{aligned} \quad (6)$$

Note that both the Lyapunov penalty and the stability penalty are quadratic functions of  $X$  in the Frobenius norm squared; in particular, they coincide when  $S = Q = I$ . Thus, the additional term  $\|XX^T\|_F^2$  is inconsequential in the design of solution algorithms. For this reason and for the ease of presentation, in what follows we omit the stability constraint in the general discussion, but comment on the modifications of algorithm when necessary.

### B. Low-Complexity Models

In several applications, it is desired to impose sparsity or low-rank structures on the state transition matrix [1], [3], [5]–[11]. In identification of gene expression networks, for example, the nonzero elements of  $A$  determine the interaction graph of the expression network [3], [5]. In this context, a sparse  $A$  is desired because one can construct a sparse network to explain data.

A common approach to promote sparsity is to impose an  $\ell_1$  constraint:

$$\|X\|_{\ell_1} := \sum_{i,j=1}^p |X_{ij}| \leq l, \quad (7)$$

where  $l$  is a prescribed positive number. Since the  $\ell_1$  norm promotes sparsity *implicitly*, the actual number of nonzeros in the solution is indirectly controlled by the threshold  $l$ . However, given a desired level of sparsity, the correct choice of  $l$  is typically not known a priori. Practical choice of  $l$  typically requires grid search or cross validation.

Alternatively, an *explicit* way to guarantee sparsity is to directly control the number of nonzeros by the cardinality constraint:

$$\text{card}(X) := \text{number of nonzero entries of } X \leq s, \quad (8)$$

where  $s$  is a positive integer. Note that the cardinality constraint is harder to deal with than the  $\ell_1$  constraint, because cardinality is nonconvex and nonsmooth. An essential ingredient of this work is the solution method that handle this difficulty.

Another approach to obtain low-complexity VAR models is to impose a low-rank constraint. A low-rank state transition matrix is useful because it implies that the data can be explained by a simpler VAR model. It is common practice to employ model reduction techniques to obtain a low-rank model as a post-processing step [15], [16].

An implicit way to promote low-rank solutions in system identification is to use a nuclear norm constraint [17]–[20]

$$\|X\|_* := \sum_{i=1}^p \sigma_i(X) \leq u, \quad (9)$$

where  $u$  is a positive number and the  $\sigma_i$ 's denote the singular values of  $X$ . Similar to the sparsity case, the threshold  $u$  may be challenging to prescribe.

Alternatively, we may impose a low-rank constraint by explicitly controlling the rank of the state transition matrix:

$$\text{rank}(X) := \text{number of nonzero singular values of } X \leq r, \quad (10)$$

where  $r$  is a positive integer. This integer may be determined from resource limitations in practice.

To summarize, we consider the estimation problem of the low-complexity VAR model:

$$\begin{aligned} \hat{A} = \underset{X \in \mathbb{R}^{p \times p}}{\text{argmin}} \quad & \frac{1}{2} \|XC - D\|_F^2 + \frac{\rho}{2} \|XSX^T + Q - S\|_F^2 \\ \text{subject to} \quad & \text{constraint (7), (8), (9), or (10)}. \end{aligned} \quad (11)$$

For convex constraints (i.e., the  $\ell_1$  and the nuclear norm), one may employ gradient projection methods, namely, taking a descent direction of the objective function and projecting onto the convex constraint sets. A gradient projection method is proposed in [1] to solve (11) with the  $\ell_1$  constraint (7). For nonconvex constraints (i.e., the cardinality and the rank constraints), we develop PALM in the subsequent section. Note that this method can also be applied to convex constraints as well.

### III. PROXIMAL ALTERNATING LINEARIZED METHOD (PALM)

In this section, we reformulate (11) into a form well suited to PALM. One advantage of PALM is its global convergence to a critical point for both convex and nonconvex constraints. We begin with replacing one of the two  $X$ 's in the quadratic Lyapunov term by a new variable  $Y$  and rewrite (11) as

$$\begin{aligned} \underset{X, Y}{\text{minimize}} \quad & \frac{1}{2} \|XC - D\|_F^2 + \frac{\rho}{2} \|YSX^T + Q - S\|_F^2 \\ \text{subject to} \quad & Y - X = 0, \\ & \text{(7), (8), (9), or (10)}. \end{aligned}$$

This change of variable will allow us to establish the Lipschitz conditions required for convergence analysis.

Let  $f$  denote the least-squares term

$$f(X) = \frac{1}{2} \|XC - D\|_F^2, \quad (12)$$

and let  $H$  denote the coupling term

$$H(X, Y) = \frac{\rho_1}{2} \|YSX^T + Q - S\|_F^2 + \frac{\rho_2}{2} \|X - Y\|_F^2, \quad (13)$$

where the penalty parameter  $\rho_1 > 0$  resumes the role of  $\rho$  in (11) and  $\rho_2 > 0$  is sufficiently large to penalize the discrepancy between  $X$  and  $Y$ . Then we obtain the following formulation

$$\underset{X, Y}{\text{minimize}} \quad \Phi(X, Y) := f(X) + g(Y) + H(X, Y), \quad (14)$$

where  $g$  is the indicator function of the constraints (7)–(10). For example,

$$g(Y) = \begin{cases} 0, & \text{card}(Y) \leq s \\ \infty, & \text{otherwise,} \end{cases} \quad (15)$$

for the cardinality constraint (8), or

$$g(Y) = \begin{cases} 0, & \text{rank}(Y) \leq r \\ \infty, & \text{otherwise,} \end{cases} \quad (16)$$

for the rank constraint (10).

#### A. Generic PALM Method

PALM alternates between computing the proximal operators of the uncoupled functions  $f$  and  $g$  around the linearization of the coupling function  $H$  at the previous iterate, hence the name [21]–[24]. To put PALM in the context of other alternating methods, suppose for the moment that  $\Phi(X, Y)$  is a *strictly* convex function. One approach for minimizing  $\Phi$  is the Gauss-Seidel iteration (also known as coordinate descent):

$$\begin{aligned} X^{k+1} &\in \underset{X}{\text{argmin}} \quad \Phi(X, Y^k) \\ Y^{k+1} &\in \underset{Y}{\text{argmin}} \quad \Phi(X^{k+1}, Y). \end{aligned}$$

For convergence, a unique solution in each minimization step is required. Otherwise, Gauss-Seidel may cycle indefinitely [25]. When  $\Phi$  is convex but *not strictly* convex, uniqueness can be achieved by including a quadratic proximal term

$$X^{k+1} \in \underset{X}{\text{argmin}} \left\{ \Phi(X, Y^k) + \frac{c_k}{2} \|X - X^k\|^2 \right\} \quad (18a)$$

$$Y^{k+1} \in \underset{Y}{\text{argmin}} \left\{ \Phi(X^{k+1}, Y) + \frac{d_k}{2} \|Y - Y^k\|^2 \right\}, \quad (18b)$$

where  $c_k$  and  $d_k$  are positive coefficients. This class of proximal methods is well studied; see [23] for a recent survey. It is worth noting that the celebrated alternating direction method of multipliers (ADMM) can be viewed as a class of proximal methods for convex problems [23].

When  $\Phi$  is nonconvex, as in our case (14), we need to modify the proximal terms to ensure convergence. As opposed to taking the proximal term around  $X^k$  as in (18a), we take the term around  $X^k$  modified with a scaled partial gradient of

$H$ :

$$X^{k+1} \in \underset{X}{\operatorname{argmin}} \left\{ f(X) + \frac{c_k}{2} \|X - U^k\|_F^2 \right\}, \quad (19)$$

where

$$U^k = X^k - \frac{1}{c_k} \nabla_X H(X^k, Y^k).$$

The parameter  $c_k$  is chosen to be greater than the Lipschitz constant of  $\nabla_X H$ . That is, if  $L_1$  satisfies

$$\|\nabla_X H(X_1, Y^k) - \nabla_X H(X_2, Y^k)\| \leq L_1(Y^k) \|X_1 - X_2\|,$$

for all  $X_1$  and  $X_2$ , then we let

$$c_k = \gamma_1 L_1(Y^k)$$

for some  $\gamma_1 > 1$ .

Similarly, we take the proximal term around  $Y^k$  modified with a scaled partial gradient of  $H$ ,

$$Y^{k+1} \in \underset{Y}{\operatorname{argmin}} \left\{ g(Y) + \frac{d_k}{2} \|Y - V^k\|_F^2 \right\}, \quad (20)$$

where

$$V^k = Y^k - \frac{1}{d_k} \nabla_Y H(X^{k+1}, Y^k).$$

The parameter  $d_k$  is given by  $d_k = \gamma_2 L_2(X^{k+1})$  for some  $\gamma_2 > 1$  where the Lipschitz constant  $L_2$  satisfies

$$\|\nabla_Y H(X^{k+1}, Y_1) - \nabla_Y H(X^{k+1}, Y_2)\| \leq L_2(X^{k+1}) \|Y_1 - Y_2\|,$$

for all  $Y_1$  and  $Y_2$ . PALM updates  $(X, Y)$  by using the iterations (19)-(20).

### B. Explicit Formulas

To implement the PALM iterations (19)-(20), one needs the Lipschitz constants  $L_1$  and  $L_2$  in order to determine the coefficients  $c_k$  and  $d_k$ , respectively. To this end, we take the partial gradients of  $H$  and obtain

$$\begin{aligned} \nabla_X H &= \rho_1(XS^T Y^T YS + (Q - S)^T YS) + \rho_2(X - Y) \\ \nabla_Y H &= \rho_1(YSX^T X S^T + (Q - S)X S^T) + \rho_2(Y - X). \end{aligned}$$

Since  $\nabla_X H$  is linear in  $X$  and  $\nabla_Y H$  is linear in  $Y$ , it follows that the Lipschitz constants admit explicit formulas

$$\begin{aligned} L_1(Y) &= \|\rho_1 S^T Y^T YS + \rho_2 I\|_F \\ L_2(X) &= \|\rho_1 SX^T X S^T + \rho_2 I\|_F. \end{aligned} \quad (21)$$

We next show that the proximal operators can be computed efficiently. The proximal operator (19) can be expressed as

$$X^{k+1} \in \underset{X}{\operatorname{argmin}} \left\{ \frac{1}{2} \|XC - D\|_F^2 + \frac{c_k}{2} \|X - U^k\|_F^2 \right\}.$$

Solving this least-squares problem yields

$$X^{k+1} = (DC^T + c_k U^k)(CC^T + c_k I)^{-1}, \quad (22)$$

where  $I$  denotes the identity matrix. When  $p > n - 1$ , one can gain some computation efficiency by inverting  $C^T C + c_k I$  instead of  $CC^T + c_k I$ , since the Woodbury formula gives equivalently,

$$X^{k+1} = (c_k^{-1} DC^T + U^k)(I - C(c_k I + C^T C)^{-1} C^T). \quad (23)$$

On the other hand, the proximal operator (20) is expressed as

$$\begin{aligned} &\underset{Y}{\operatorname{minimize}} \quad \frac{d_k}{2} \|Y - V^k\|_F^2 \\ &\text{subject to} \quad (7), (8), (9), \text{ or } (10). \end{aligned}$$

For the cardinality constraint (8), the optimal  $Y$  is obtained by keeping the  $s$  largest elements of  $V^k$  in magnitude and zero out the rest of the elements in  $V^k$ . This is because the Frobenius norm squared is the squared sum of the elements of  $Y - V^k$ . For the rank constraint (10), by the Eckart–Young theorem, the optimal  $Y$  is the best rank- $r$  approximation of  $V^k$  obtained by the truncated SVD.

For the  $\ell_1$  constraint (7), the projection onto the  $\ell_1$ -ball can be computed by an algorithm developed in [26]. For the nuclear norm constraint (9), the optimal solution  $Y$  can be computed by performing the singular value decomposition of  $V^k$  and then projecting the singular values of  $V^k$  onto the  $\ell_1$ -ball.

With these details, we summarize the computational steps in Algorithm 1. Two remarks follow.

---

**Algorithm 1** PALM for problem (14) with cardinality constraint (15) or rank constraint (16).

---

Initialization: Start with any  $(X^0, Y^0)$ .

**for**  $k = 0, 1, 2, \dots$  **until** convergence **do**

    // Update  $X^{k+1}$

    Compute the Lipschitz constant

$$L_1(Y^k) = \|\rho_1 S^T Y^{kT} Y^k S + \rho_2 I\|_F.$$

    Compute  $c_k = \gamma_1 L_1(Y^k)$  for some  $\gamma_1 > 1$ .

    Compute partial gradient

$$\begin{aligned} \nabla_X H(X^k, Y^k) &= \rho_1(X^k S^T Y^{kT} Y^k S \\ &\quad + (Q - S)^T Y^k S) + \rho_2(X^k - Y^k). \end{aligned}$$

    Update the proximal point

$$U^k = X^k - \frac{1}{c_k} \nabla_X H(X^k, Y^k).$$

**if**  $p \leq n - 1$  **then**

$$X^{k+1} = (DC^T + c_k U^k)(CC^T + c_k I)^{-1}$$

**else**

$$X^{k+1} = (c_k^{-1} DC^T + U^k)(I - C(c_k I + C^T C)^{-1} C^T).$$

**end if**

    // Update  $Y^{k+1}$

    Compute the Lipschitz constant

$$L_2(X^{k+1}) = \|\rho_1 S X^{(k+1)T} X^{k+1} S^T + \rho_2 I\|_F.$$

    Compute  $d_k = \gamma_2 L_2(X^{k+1})$  for some  $\gamma_2 > 1$ .

    Compute partial gradient

$$\begin{aligned} \nabla_Y H(X^{k+1}, Y^k) &= \rho_1(Y^k S X^{(k+1)T} X^{k+1} S^T \\ &\quad + (Q - S)X^{k+1} S^T) + \rho_2(Y^k - X^{k+1}). \end{aligned}$$

    Update the proximal point

$$V^k = Y^k - \frac{1}{d_k} \nabla_Y H(X^{k+1}, Y^k).$$

**if**  $g$  is the cardinality constraint (8) **then**

$Y^{k+1} = \mathcal{I}_s \circ V^k$ , where  $(\mathcal{I}_s)_{ij} = 1$  if  $(|V^k|)_{ij}$  is greater than or equal to the  $s$ -th largest element of  $|V^k|$ , and  $(\mathcal{I}_s)_{ij} = 0$  otherwise.

**else if**  $g$  is the rank constraint (10) **then**

$Y^{k+1}$  is the rank- $r$  truncated SVD of  $V^k$ .

**end if**

**end for**

---

**Remark 1** (Stepsize). *Typical descent-type methods (e.g., gradient projection) require an appropriate stepsize to ensure sufficient decrease in the objective value. In contrast, PALM needs no stepsize rule. We will see in the convergence analysis in Section IV that the objective function is guaranteed to decrease sufficiently. This feature provides computational advantage of PALM over descent-type algorithms, when the computation of the projection is nontrivial.*

**Remark 2** (Stability). *As discussed in Section II-A, we can incorporate the stability constraint by penalizing  $\|XX^T\|_F^2$  in the cost function. In this case, the coupling term becomes*

$$H(X, Y) = \frac{\rho_1}{2} \|YSX^T + Q - S\|_F^2 + \frac{\rho_2}{2} \|X - Y\|_F^2 + \frac{\mu}{2} \|YX^T\|_F^2.$$

*Its partial gradients are given by*

$$\nabla_X H = \rho_1(XS^TY^TY S + (Q - S)^TY S) + \rho_2(X - Y) + \mu XY^TY$$

$$\nabla_Y H = \rho_1(YSX^TXS^T + (Q - S)XS^T) + \rho_2(Y - X) + \mu YX^TX,$$

*and the Lipschitz constants are*

$$L_1(Y) = \|\rho_1 S^TY^TY S + \mu Y^TY + \rho_2 I\|_F$$

$$L_2(X) = \|\rho_1 SX^TXS^T + \mu X^TX + \rho_2 I\|_F.$$

*With these modifications, Algorithm 1 is directly applied to the problem (6) subject to low-complexity constraints.*

#### IV. CONVERGENCE ANALYSIS

In this section, we show that Algorithm 1 globally converges to a critical point of the nonconvex, nonsmooth problem (14). Furthermore, the objective value is monotonically decreasing. Our analysis builds upon the seminal work in [24] for the global convergence of the generic PALM algorithm. Our contributions are the establishments of Lipschitz conditions and the KL property of (14). In particular, the Lipschitz conditions ensure sufficient decrease of the objective value and the KL property is the key step for the convergence to a critical point.

We begin with a technical lemma on the Lipschitz conditions of  $\Phi$ .

**Lemma 1.** *The objective function  $\Phi$  in (14) satisfies the following properties:*

- 1)  $\inf_{X,Y} \Phi(X, Y) > -\infty$ ,  $\inf_X f(X) > -\infty$ , and  $\inf_Y g(Y) > -\infty$ .
- 2) *For a fixed  $Y$ , the partial gradient  $\nabla_X H(X, Y)$  is globally Lipschitz, that is,*

$$\|\nabla_X H(X_1, Y) - \nabla_X H(X_2, Y)\| \leq L_1(Y) \|X_1 - X_2\|$$

*for all  $X_1$  and  $X_2$ . Likewise, for a fixed  $X$ , the partial gradient  $\nabla_Y H(X, Y)$  is globally Lipschitz, that is,*

$$\|\nabla_Y H(X, Y_1) - \nabla_Y H(X, Y_2)\| \leq L_2(X) \|Y_1 - Y_2\|$$

*for all  $Y_1$  and  $Y_2$ .*

- 3) *There exist bounded constants  $q_1^-, q_1^+, q_2^-, q_2^+ > 0$  such that*

$$\inf_k \{L_1(Y^k)\} \geq q_1^- \quad \text{and} \quad \inf_k \{L_2(X^k)\} \geq q_2^-$$

$$\sup_k \{L_1(Y^k)\} \leq q_1^+ \quad \text{and} \quad \sup_k \{L_2(X^k)\} \leq q_2^+. \quad (24)$$

- 4) *The entire gradient  $\nabla H(X, Y)$  is Lipschitz continuous on the bounded subsets of  $\mathbb{R}^{p \times p} \times \mathbb{R}^{p \times p}$ .*

A few comments are in order. Property 1) ensures that each proximal operator in PALM is well defined and the minimization of  $\Phi$  is also well defined. Property 2) on the boundedness of the Lipschitz constants is critical for convergence. Note that the block-Lipschitz property in  $X$  and  $Y$  is weaker than standard assumptions in proximal methods that require  $\Phi$  to be globally Lipschitz in *joint* variables  $(X, Y)$ . Property 3) guarantees that the Lipschitz constants for the partial gradients are lower and upper bounded by finite numbers. Property 4) is a technical condition for controlling the distance between two consecutive steps in the sequence  $(X^k, Y^k)$ .

*Proof.* Property 1) is a direct consequence of the nonnegativity of  $f$  in (12),  $H$  in (13), and the indicator function  $g$  in (15) and (16). Property 2) follows from the Lipschitz constants derived in (21). To show property 3), note that  $L_1(Y)$  in (21) is clearly bounded below for all  $Y$ , in particular,

$$L_1^2(Y) = \rho_1^2 \|S^TY^TY S\|_F^2 + 2\rho_1\rho_2 \|YS\|_F^2 + \rho_2^2 \geq \rho_2^2.$$

On the other hand, since  $Y^k$  is the minimizer of a feasible problem over a bounded set, it is bounded for all  $k$  and hence  $L_1(Y^k)$  is bounded above. Thus, the entire sequence  $L_1(Y^k)$  satisfies the upper and lower bounds in (24). An analogous argument shows that the Lipschitz constant  $L_2(X)$  satisfies (24). Property 4) is a direct consequence of the twice continuous differentiability of  $H$  and the mean value theorem.  $\square$

Applying Lemma 1, it follows that the objective value is monotonically decreasing. Specifically, we have the following result.

**Proposition 1.** *Let  $Z^k := (X^k, Y^k)$  be a sequence generated by Algorithm 1. Then the following result holds*

$$\frac{\delta}{2} \|Z^{k+1} - Z^k\|_F^2 < \Phi(Z^k) - \Phi(Z^{k+1}), \quad \forall k \geq 0$$

where  $\delta = \min\{(\gamma_1 - 1)q_1^-, (\gamma_2 - 1)q_2^-\}$ .

*Proof.* Apply our Lemma 1 to the Lemma 3 of [24].  $\square$

Note that  $\delta > 0$  throughout iterations because  $\gamma_1, \gamma_2 > 1$  (see Algorithm 1) and  $q_1^-, q_2^- > 0$  (see Lemma 1). Thus, the convergence of the decision variable  $Z^k$  can be measured by the convergence of the objective value. The numerical experiments in Section V verify this convergence behavior.

We next show that Algorithm 1 converges to a critical point of  $\Phi$ . We need a few definitions.

**Definition 1** (KL property). *Let  $\mathbf{f} : \mathbb{R}^d \rightarrow (-\infty, +\infty]$  be proper and lower semicontinuous. The function  $\mathbf{f}$  is said to have the Kurdyka-Lojasiewicz (KL) property at  $\bar{u} \in$*

$\text{dom } \partial \mathbf{f} := \{u \in \mathbb{R}^d : \partial \mathbf{f}(u) \neq \emptyset\}$  if there exist  $\eta \in (0, +\infty]$ , a neighborhood  $\mathcal{N}$  of  $\bar{u}$ , and a function  $\psi$  such that for all

$$u \in \mathcal{N} \cap \{\mathbf{f}(\bar{u}) < \mathbf{f}(u) < \mathbf{f}(\bar{u}) + \eta\},$$

the following inequality holds:

$$\psi'(\mathbf{f}(u) - \mathbf{f}(\bar{u})) \cdot \text{dist}(0, \partial \mathbf{f}(u)) \geq 1,$$

where  $\text{dist}(x, s) := \inf\{\|y - x\| : y \in s\}$  denotes the distance from a point  $x \in \mathbb{R}^d$  to a set  $s \subset \mathbb{R}^d$ . A function  $\mathbf{f}$  is called a KL function if  $\mathbf{f}$  satisfies the KL property at each point of the domain of the gradient  $\partial \mathbf{f}$ .

**Definition 2** (Semialgebraic function). A subset  $\mathcal{S}$  of  $\mathbb{R}^d$  is a real semi-algebraic set if there exists a finite number of real polynomial functions  $\mathbf{g}_{ij}$  and  $\mathbf{h}_{ij} : \mathbb{R}^d \rightarrow \mathbb{R}$  such that

$$\mathcal{S} = \bigcup_{j=1}^p \bigcap_{i=1}^q \{u \in \mathbb{R}^d : \mathbf{g}_{ij}(u) = 0 \text{ and } \mathbf{h}_{ij}(u) < 0\}.$$

A function  $\mathbf{h} : \mathbb{R}^d \rightarrow (-\infty, +\infty]$  is called semi-algebraic function if its graph  $\{(u, v) \in \mathbb{R}^{d+1} : \mathbf{h}(u) = v\}$  is a semi-algebraic subset of  $\mathbb{R}^{d+1}$ .

It is known that a proper, lower semicontinuous, and semi-algebraic function satisfies the KL property (see Theorem 3 of [24]). Based on this relation, we show that  $\Phi$  is such a function.

**Lemma 2.** The objective function  $\Phi$  in (14) satisfies the KL property.

*Proof.* Since  $\Phi$  is the summation of smooth functions  $f$ ,  $H$  and the indicator function  $g$  that is lower semicontinuous, it follows that  $\Phi$  is a proper and lower semicontinuous function. To see that  $\Phi$  is semi-algebraic, let us examine each term. Clearly,  $f$  and  $H$  are semi-algebraic because they are real-valued polynomials. Moreover, the indicator function of the semi-algebraic set  $\{Y | \text{card}(Y) \leq s\}$  is semi-algebraic, and the indicator function of the semi-algebraic set  $\{Y | \text{rank}(Y) \leq r\}$  is also semi-algebraic; see Examples 2 and 3 of [24]. Then, a finite sum of semi-algebraic functions is semi-algebraic, which concludes the proof.  $\square$

The KL property of  $\Phi$  established in Lemma 2 allows us to invoke the convergence results in [24].

**Proposition 2.** Let  $Z^k = (X^k, Y^k)$  be a sequence generated by Algorithm 1. The following results hold.

- 1) The sequence  $\{Z^k\}$  has a finite length, that is,

$$\sum_{k=1}^{\infty} \|Z^{k+1} - Z^k\|_F < \infty.$$

- 2) The sequence  $\{Z^k\}$  converges to a critical point  $Z^* = (X^*, Y^*)$  of  $\Phi$ .

*Proof.* Apply our Lemmas 1 and 2 to the Theorem 1 of [24].  $\square$

## V. NUMERICAL EXPERIMENTS

In this section, we evaluate the performance of PALM on synthetic data and real-world data. We demonstrate that the

PALM solution converges to a matrix with the prescribed level of sparsity or rank. Furthermore, the objective value decreases monotonically as predicted by the convergence analysis.

We also compare the estimation errors for VAR models obtained from nonconvex and convex constraints; in particular, we focus on the cardinality constraint and the  $\ell_1$  constraint. Our numerical results show that the cardinality constraint achieves a smaller estimation error than the  $\ell_1$  constraint on a variety of systems drawn from the COMPElib library [27], [28], if both constraints are handled by PALM. Moreover, we compare the PALM algorithm with gradient projection and show that the former outperforms the latter for handling the  $\ell_1$  constraint.

In our experiments, we assume  $Q = \sigma^2 I$  for the covariance matrix of  $\epsilon(t)$  in (1). We set  $\gamma_1 = \gamma_2 = 2$  in Algorithm 1. The hyperparameters  $\rho_1$  and  $\sigma$  are determined through cross validation.

### A. Synthetic Data

We test the performance of PALM on a sparse example and a low-rank example with the matrix size  $200 \times 200$ . In both examples, we use time series of length  $n = 50$  for training and  $m = 800$  for testing. For the length of steady-state data,  $N = 1600$ . The performance of the identified VAR model is evaluated by using the normalized error and the cosine score proposed in [1]

$$\text{Normalized error: } \frac{1}{m-1} \sum_{t=1}^{m-1} \frac{\|x(t+1) - \hat{A}x(t)\|}{\|x(t+1) - x(t)\|}$$

Cosine score:

$$\frac{1}{m-1} \sum_{t=1}^{m-1} \frac{|(x(t+1) - x(t))^T (x(t) - \hat{A}x(t))|}{\|x(t+1) - x(t)\| \|x(t) - \hat{A}x(t)\|}.$$

A smaller normalized error (lower bounded by 0) and a higher cosine score (upper bounded by 1) imply better performance.

1) *Sparse Example:* The sparse matrix is generated by using the rule  $A = (0.95M) / \max_k(|\lambda_k(M)|)$ , where  $M$  has 5000 normally distributed nonzeros and  $\lambda_k(M)$  denotes the eigenvalues of  $M$ . We set  $s = 5000$  in the cardinality constraint (8).

Figure 1 shows the convergence results. The objective value monotonically decreases, as Proposition 1 indicates. The errors in two consecutive PALM steps, namely,

$$\begin{aligned} e_X^k &= \|X^{k+1} - X^k\|_F, & e_Y^k &= \|Y^{k+1} - Y^k\|_F, \\ e_{XY}^k &= \|X^k - Y^k\|_F, \end{aligned}$$

all decrease quickly. It takes PALM fewer than 300 iterations to reach  $e_X, e_Y \leq 10^{-5}$  and  $e_{XY} \leq 2 \times 10^{-3}$ . Note that the solution has exactly 5000 nonzeros as required by the cardinality constraint. For the estimated matrix  $\hat{A}$ , the normalized error is 0.3268 and the cosine score is 0.9447.

2) *Low-Rank Example:* The low-rank matrix is generated by using the rule  $A = \mathcal{U}\Sigma\mathcal{V}$ , where  $\Sigma \in \mathbb{R}^{25 \times 25}$  is a diagonal matrix with random diagonal entries uniformly distributed in  $[0, 1)$ , and  $\mathcal{U} \in \mathbb{R}^{200 \times 25}$  and  $\mathcal{V} \in \mathbb{R}^{25 \times 200}$  are random orthonormal matrices. By construction  $A \in \mathbb{R}^{200 \times 200}$  is stable

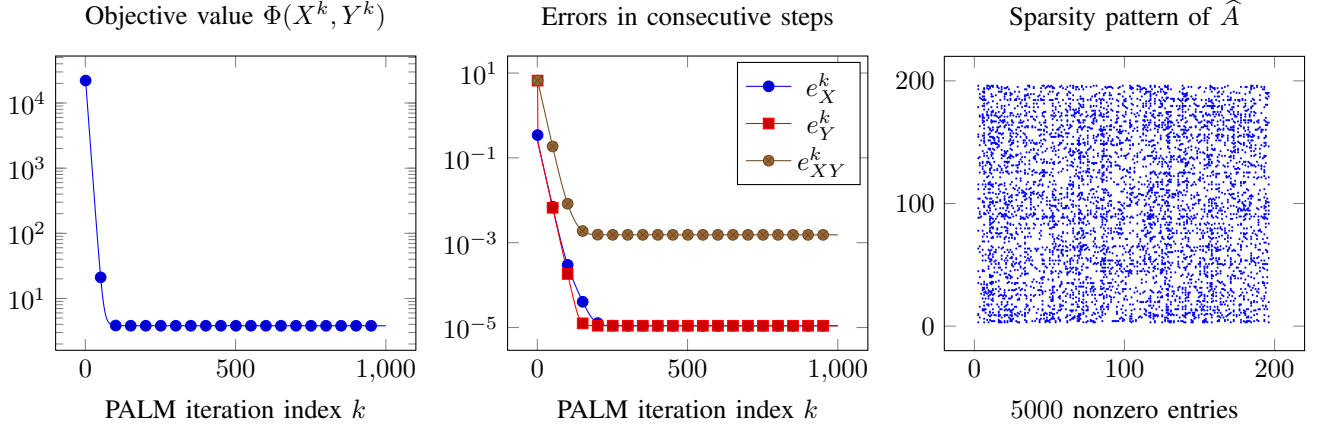


Fig. 1: Convergence results of PALM for the sparse example: the objective value (left), the errors in consecutive steps (middle), and the sparse solution with 5000 nonzero entries (right).

with  $\text{rank}(A) = 25$ . Thus we set  $r = 25$  in the rank constraint (10).

Figure 2 shows the convergence results. Similar to those for the sparse example in Figure 1, we observe that the objective value  $\Phi$  monotonically decreases and the errors in two consecutive PALM steps decrease quickly. It takes PALM less than 300 iterations to reach  $e_X, e_Y \leq 2 \times 10^{-6}$  and  $e_{XY} \leq 2 \times 10^{-4}$ . The solution has a numerical rank 25, as required by the rank constraint. For the estimated matrix  $\hat{A}$ , the normalized error is 0.6977 and the cosine score is 0.7164.

### B. Electricity Load Data

We explore the utility of the low-complexity models on an electricity load data set from the UCI repository.<sup>1</sup> The data set consists of 15-minute interval load readings of clients over a few years. To investigate the daily dynamics, we aggregate the data in every 24-hour interval. Because over half of the clients are not registered in the first year, we start from the second-year data and collect clients whose time series are uninterrupted. Interruptions may arise from late registration of clients, missing data, or a period of low electricity consumption due to inactivity. Such a preprocessing results in 272 clients and 1095 daily readings per client. We further subtract each time series by its seasonal mean, that is, the mean of the same day along all the years, and normalize it by the standard deviation.

We first use the least-squares estimator (3) to obtain a baseline model. To this end, the first 995 days are used for training and the last 100 days are used for testing. Next, we test the low-complexity models (11) with different thresholds for the cardinality and the rank constraint. In particular, we set  $s \in \{100p, 125p, 150p, 175p, 200p, p^2\}$  and similarly  $r \in \{100, 125, 150, 175, 200, p\}$ . We take the first  $n$  days with  $n \in \{25, 50, 75, 100\}$  as the training data, the last 100 days as the testing data, and 600 randomly sampled days in the remaining dataset as the steady-state data. We repeat the experiment five times for each set of  $s$ ,  $r$ , and  $n$ .

Figure 3 shows the performance measures as the number  $n$  of training data and the sparsity level  $s$  vary. Three observations are made. First, the results are not sensitive to the length of the training data, because the performance varies slightly with  $n$ . In particular, the curves are approximately horizontal. Second, as the complexity of the transition matrix increases (e.g., a larger  $s$ ), the performance gets closer to that of the least-squares estimator. Third, in the case of no constraints (i.e.,  $s = p^2$ ), the Lyapunov-penalized VAR model (4) performs as well as the least-squares estimator (3). In other words, the Lyapunov-penalized VAR model with a small number of time sequence data and a large number of nonsequence data is as competitive as the least-squares estimator with a large amount of time sequence data. This result demonstrates the utility of the proposed method when time sequence data is limited.

The error bars in Fig. 3, due to random sampling of steady-state data, are much narrower than the size of the markers and hence they are not visible. The results for the rank constraints are similar to Fig. 3 and hence they are omitted for brevity.

### C. Comparison between nonconvex and convex constraints, PALM and gradient projection

As mentioned earlier, PALM can handle both nonconvex constraints (i.e., (8) and (10)) and convex constraints (i.e., (7) and (9)). It is thus of interest to compare the performance of both classes of constraints in identifying low-complexity models. We test sparse models with cardinality and  $\ell_1$  constraints. When the  $\ell_1$  constraint is employed, we also compare the performance of PALM and that of the gradient projection method.

We test a variety of systems from the *COMpleib* library [27], [28] that includes applications from aircraft, helicopter, jet engine, reactor, decentralized interconnected systems, and wind energy. The set consists of 40 continuous-time systems with the dimension of the state matrix  $A_c \in \mathbb{R}^{p \times p}$  ranging from  $p = 3$  to  $p = 40$ . We generate the discrete-time state transition matrix by taking  $A = \exp(A_c \Delta t)$  where  $\Delta t = 1$ . In the resulting 40 discrete-time models, 25 of

<sup>1</sup><http://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014>

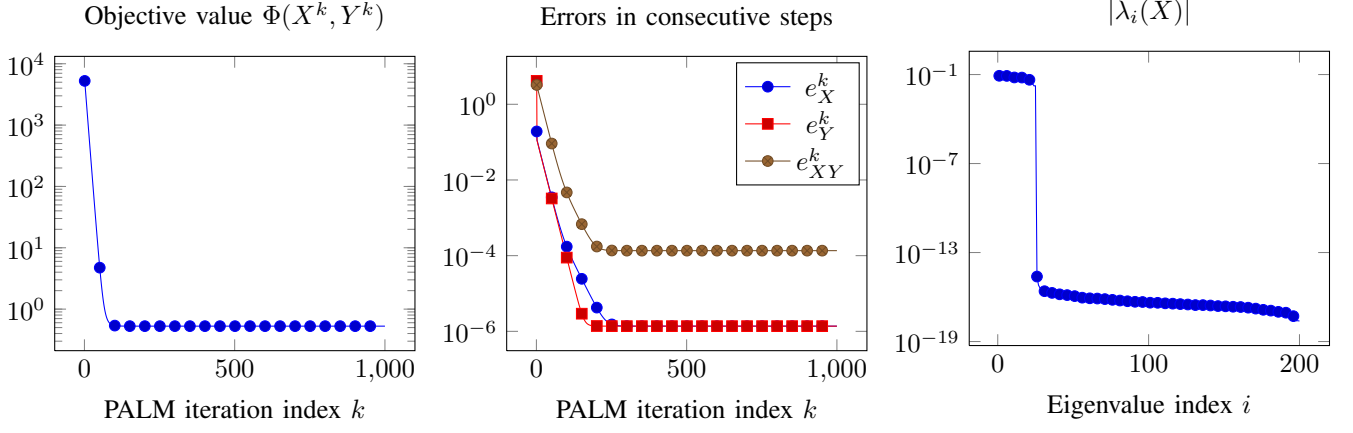


Fig. 2: Convergence results of PALM for the low-rank example: the objective value (left), the errors in consecutive steps (middle), and the low-rank solution with 25 nonzero eigenvalues (right).

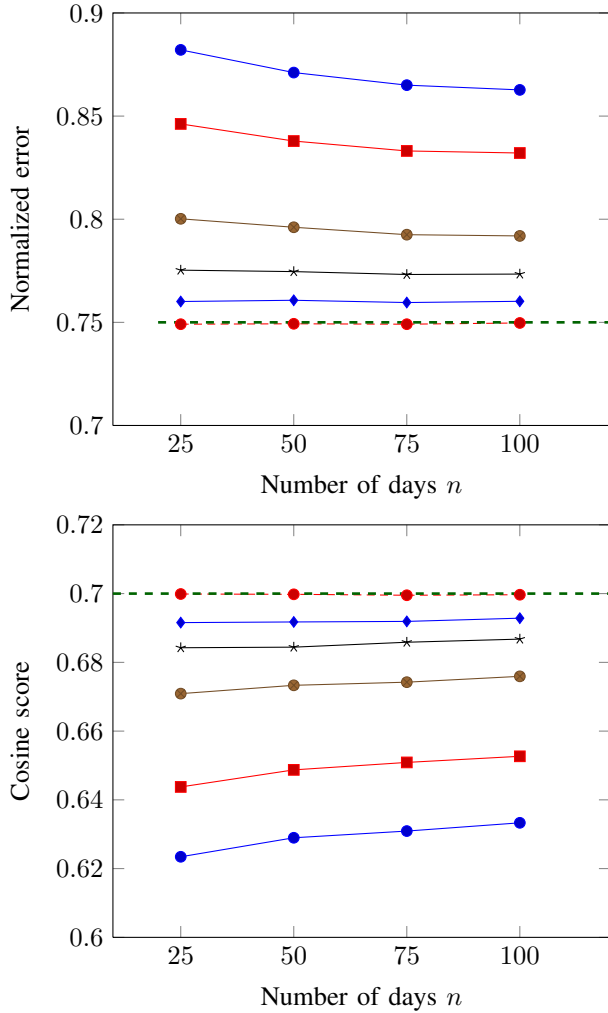


Fig. 3: Electricity load data: Performance comparison between the least-squares estimator (3) when  $p < n$  (—) and the Lyapunov-penalized model (11) when  $p > n$ , with different levels of cardinality:  $s = 100p$  ( $\bullet$ ),  $s = 125p$  ( $\blacksquare$ ),  $s = 150p$  ( $\bullet$ ),  $s = 175p$  ( $*$ ),  $s = 200p$  ( $\diamond$ ),  $s = p^2$  ( $\circ$ ).

them are unstable. To get stable VAR models, we scale the transition matrix  $A \leftarrow A/(2\tau)$  where  $\tau$  is the spectral radius of  $A$ . For each model, we generate  $n = p/2$  and  $m = p$  time sequence data for training and testing, respectively, and  $N = 5n$  nonsequence data.

We use PALM to solve problem (11) with the cardinality constraint (8) and the  $\ell_1$  constraint (7). For a fair comparison, the number of nonzero elements of the solution  $\hat{A}$  must be the same in both cases. For the cardinality constraint, we set the desired number of nonzeros to be  $s = \alpha p^2$  for  $\alpha \in \{1/2, 1/4, 1/8\}$ . For the  $\ell_1$  constraint  $\|X\|_{\ell_1} \leq l$ , the upper bound  $l$  that yields the desired number of nonzeros is not known a priori. To find the matching  $l$ , we use a simple bisection method: Starting from an interval  $[l_{\text{low}}, l_{\text{up}}]$  that contains the unknown  $l$ , repeatedly solve (11) and divide the interval by half, until the desired  $l$  is found or the interval is sufficiently small.

Additionally, we use gradient projection (GP) to solve (11) with the  $\ell_1$  constraint.

Table I shows the performance of PALM-card, PALM- $\ell_1$ , and GP- $\ell_1$  methods. PALM-card outperforms the other two approaches in achieving a smaller normalized error and a higher cosine score. The percentage of cases that PALM-card outperforms the others increases with the level of sparsity, from 62.5% for  $\alpha = 1/4$  to 82.5% for  $\alpha = 1/8$ . Similarly PALM-card yields the highest cosine score in 60% of the systems when  $\alpha = 1/4$  and in 77.5% of the systems when  $\alpha = 1/8$ . When the  $\ell_1$  constraint is used, PALM outperforms GP when  $\alpha = 1/2$  and  $\alpha = 1/4$ .

Figure 4 shows the normalized error and the cosine score for the three methods when  $\alpha = 1/4$ . Note that for 14 test problems, the errors resulted from GP- $\ell_1$  is at least two times (and up to 43 times) of the errors from PALM-card and PALM- $\ell_1$ . Similar observations can be made for the cosine score. For 12 test problems, PALM-card and PALM- $\ell_1$  result in cosine scores that are at least twice of those obtained from GP- $\ell_1$ . These results suggest that (i) the cardinality constraint is more effective than the  $\ell_1$  constraint and (ii) PALM outperforms GP by converging to better solutions.

Finally, we comment on the computational efficiency of



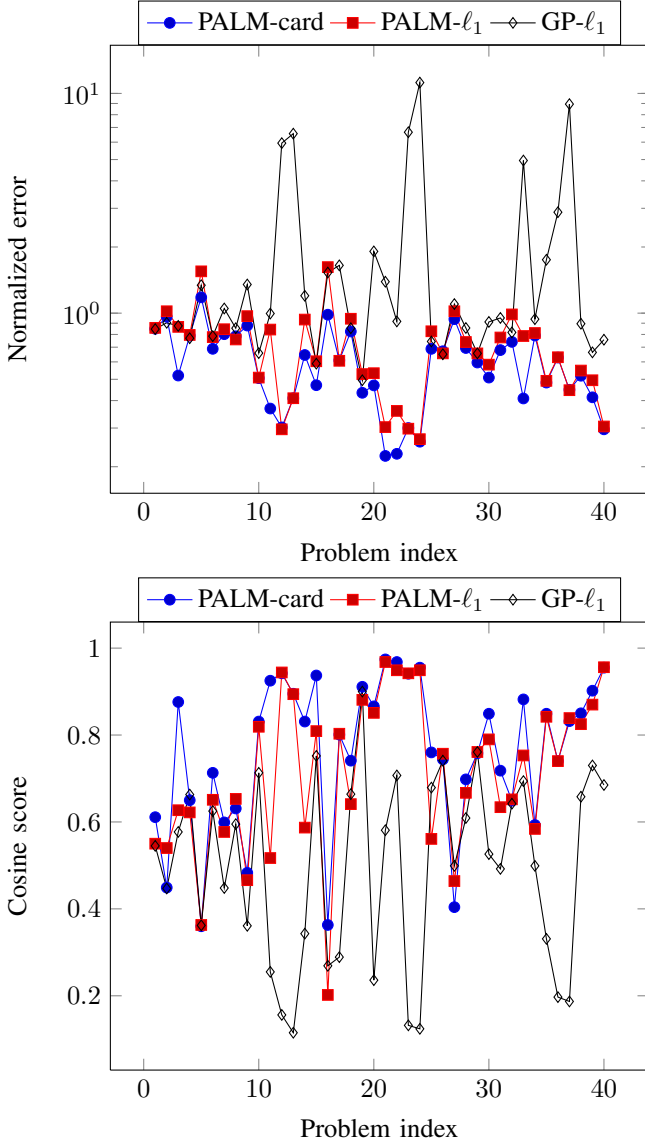


Fig. 4: Performance of PALM-card, PALM- $\ell_1$ , and GP- $\ell_1$  on COMPLEIB test problems, with sparsity level  $\alpha = 1/4$ .

the two methods. As mentioned earlier, PALM requires no tuning for the stepsize but GP does. This feature makes PALM computationally more efficient when the projection onto the constraint set becomes nontrivial. For the projection onto the  $\ell_1$ -ball, it turns out that the most time-consuming computation in GP is to compute the stepsize by using the Armijo rule along the projection-arc [29]. This is because GP requires a number of  $\ell_1$  projections to compute the stepsize. As a consequence, PALM is computationally more efficient than GP for the  $\ell_1$  constraint.

## VI. CONCLUSIONS

In this paper, we formulate a constrained optimization problem for estimating the state transition matrix of a vector autoregressive model, with limited time sequence data but abundant nonsequence steady-state data. To reduce the complexity of the model, we propose imposing a cardinality or a

rank constraint on the transition matrix. We develop a PALM algorithm to solve the resulting nonconvex, nonsmooth problem and establish its global convergence through verifying the crucial technical conditions required by PALM. Our numerical experiments verify the convergence theory and demonstrate the effectiveness of the developed method.

Several directions may be pursued following this work. First, we observe a linear convergence of the algorithm in practice (e.g., Fig. 1 and Fig. 2). It is thus of interest to investigate the convergence rate of PALM in a theoretical setting. Second, the identified model is only one of many legitimate models that explain the given data, depending on formulation. One wants to understand under what conditions the proposed model is asymptotically consistent with the ground truth. Finally, although the model itself has a low complexity, the optimization still requires the storage and the computation with  $p \times p$  matrices. It would be interesting to investigate methods (e.g., randomized SVD) that approximately update the unknowns and render the optimization also of low complexity.

## ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract number DE-AC02-06CH11357. J. Chen is supported in part by XDATA program of the Defense Advanced Research Projects Agency (DARPA), administered through Air Force Research Laboratory contract FA8750-12-C-0323.

## REFERENCES

- [1] T.-K. Huang and J. G. Schneider, "Learning auto-regressive models from sequence and non-sequence data," in *Advances in Neural Information Processing Systems*, 2011, pp. 1548–1556.
- [2] R. Yoshida, S. Imoto, and T. Higuchi, "Estimating time-dependent gene networks from time series microarray data by dynamic linear models with Markov switching," in *Proceedings of the 2005 Computational Systems Bioinformatics Conference*, 2005, pp. 289–298.
- [3] M. M. Zavlanos, A. A. Julius, S. P. Boyd, and G. J. Pappas, "Inferring stable genetic networks from steady-state data," *Automatica*, vol. 47, no. 6, pp. 1113–1122, 2011.
- [4] Y. K. Wang, D. G. Hurley, S. Schnell, E. J. Crampin *et al.*, "Integration of steady-state and temporal gene expression data for the inference of gene regulatory networks," *PloS one*, vol. 8, no. 8, p. e72103, 2013.
- [5] A. Fujita, J. R. Sato, H. M. Garay-Malpartida, R. Yamaguchi, S. Miyano, M. C. Sogayar, and C. E. Ferreira, "Modeling gene expression regulatory networks with the sparse vector autoregressive model," *BMC Systems Biology*, vol. 1, no. 1, p. 39, 2007.
- [6] H. Wang, G. Li, and C.-L. Tsai, "Regression coefficient and autoregressive order shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 1, pp. 63–78, 2007.
- [7] A. Gupta and Z. Bar-Joseph, "Extracting dynamics from static cancer expression data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 5, pp. 172–182, 2008.
- [8] T.-K. Huang and J. Schneider, "Learning linear dynamical systems without sequence information," in *Proceedings of the 26th International Conference on Machine Learning*, 2009, pp. 425–432.
- [9] F. Han and H. Liu, "Transition matrix estimation in high dimensional time series," in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 172–180.
- [10] M. T. Bahadori, Y. Liu, and E. P. Xing, "Fast structure learning in generalized stochastic processes with latent factors," in *Proceedings of the 19th SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 284–292.

TABLE I: Performance of PALM-card, PALM- $\ell_1$ , and GP- $\ell_1$  on COMPLEIB test problems. The sparsity level is indicated by  $\alpha = s/p^2$ . The table shows the number of test problems where each method outperforms the other two. For example, when  $\alpha = 1/4$ , PALM-card achieves the smallest normalized error in 30 test problems and the highest cosine score in 27 test problems.

$\alpha$	Normalized error			Cosine score		
	PALM-card	PALM- $\ell_1$	GP- $\ell_1$	PALM-card	PALM- $\ell_1$	GP- $\ell_1$
1/2	25	12	3	24	12	4
1/4	30	6	4	27	10	3
1/8	33	2	5	31	3	6

- [11] P. Geiger, K. Zhang, B. Schoelkopf, M. Gong, and D. Janzing, "Causal inference by identification of vector autoregressive processes with hidden components," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 1917–1925.
- [12] T.-K. Huang, L. Song, and J. Schneider, "Learning nonlinear dynamic models from nonsequenced data," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 2010, pp. 350–357.
- [13] J. E. Larvie, M. S. Gorji, and A. Homaifar, "Inferring stable gene regulatory networks from steady-state data," in *41st Annual Northeast Biomedical Engineering Conference*, 2015, pp. 1–2.
- [14] A. Lozano, N. Abe, Y. Liu, and S. Rosset, "Grouped graphical granger modeling for gene expression regulatory networks discovery," *Bioinformatics*, vol. 25, no. 12, pp. i110–i118, 2009.
- [15] M. Verhaegen, "Identification of the deterministic part of MIMO state space models given in innovations form from input-output data," *Automatica*, vol. 30, no. 1, pp. 61–74, 1994.
- [16] L. Ljung, "System identification," in *Signal Analysis and Prediction*. Springer, 1998, pp. 163–173.
- [17] M. Fazel, H. Hindi, and S. P. Boyd, "A rank minimization heuristic with application to minimum order system approximation," in *Proceedings of the 2001 American Control Conference*, 2001, pp. 4734–4739.
- [18] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1235–1256, 2009.
- [19] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.
- [20] M. Fazel, T. K. Pong, D. Sun, and P. Tseng, "Hankel matrix rank minimization with applications to system identification and realization," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 946–977, 2013.
- [21] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality," *Mathematics of Operations Research*, vol. 35, no. 2, pp. 438–457, 2010.
- [22] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods," *Mathematical Programming*, vol. 137, no. 1-2, pp. 91–129, 2013.
- [23] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.
- [24] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.
- [25] M. J. D. Powell, "On search directions for minimization algorithms," *Mathematical Programming*, vol. 4, no. 1, pp. 193–201, 1973.
- [26] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the  $\ell_1$ -ball for learning in high dimensions," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 272–279.
- [27] F. Leibfritz, "Compleib: Constraint matrix optimization problem library - A collection of test examples for nonlinear semidefinite programs, control system design and related problems," University of Trier, Tech. Rep., 2004.
- [28] F. Leibfritz and W. Lipinski, "COMPLEIB 1.0 - user manual and quick reference," University of Trier, Tech. Rep., 2004.
- [29] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific Belmont, 1999.